Trust Issues with Opportunistic Encryption

Scott Rose, Stephen Nightingale, Doug Montgomery
{scottr, night, dougm@nist.gov}
National Institute of Standards and Technology (NIST)[1]
January 15, 2014 updated March 10, 2014

Abstract

Recent revelations have shed light on the ease and potential of eavesdropping on Internet traffic; raising concerns of privacy for almost every Internet user. In response, protocol designers and network operators have begun deploying encryption (often opportunistic) to protect the confidentiality of users' communications. The lack of authentication in opportunistic encryption could have the perverse affect of putting more end users at risk: thinking that they are "secure", an end user may divulge private information to an imposter instead of the service they believe they have contacted. When adding protection mechanisms to protocols, designers and implementers should not downplay the importance of authentication in order to make opportunistic encryption easier to deploy.

Introduction

During the 88th Internet Engineering Task Force (IETF) meeting in November 2013, there was a technical plenary addressing how the IETF, as a group, should address pervasive monitoring of Internet traffic. It was decided that the IETF would pursue privacy-enhancing extensions to existing protocols and urge the implementation and use of encryption for all Internet traffic [1]. This has lead to a push to include the use of opportunistic encryption to provide for the confidentiality of traffic in protocols that traditionally either did not have the option, or the option was not widely used (e.g. Simple Mail Transfer Protocol [SMTP]).

Opportunistic encryption usually refers to having the option to encrypt the communication between two parties without requiring or providing authentication of either party, as there are often no a priori arrangements that make it easy, or even possible to perform. In other words, providing confidentiality (stopping eavesdroppers) but not providing authentication. For example, having two Mail Transfer Agents (MTA's) exchange email messages via SMTP over a Transport Layer Security (TLS) connection protects the contents of the email, but usually lacks authentication of the certificate presented during the TLS handshake. End users also commonly encounter using an opportunistic encrypted channel when using a web browser and seeing a HTTPS certificate warning. When users click the "Ok" (or similar) button to proceed with viewing the webpage, they are agreeing to use a connection that is not authenticated, but encrypted.

---

[1] *The views presented in this whitepaper are those of the authors and do not necessarily represent the views or policies of NIST*

In many scenarios, this level of security is acceptable and should be encouraged to limit third party eavesdropping. However, in other scenarios, end users would like to have some level of authentication that the party they believe they are communicating with is, in fact, the party they are actually communicating with. Most people would not be concerned with authentication when viewing a webpage with movie times, sports scores, etc. where little of their private information is shared, but when checking their credit scores, bank balance or medical records, most end users would like some assurance that they are communicating with their actual bank, doctor, etc. and not an imposter. Authentication is seen as an important component to most security protocols, but with the current pressure to combat pervasive monitoring on the Internet there is a concern that authentication will be deferred in favor of opportunistic encryption in a rushed deployment. There are alternatives to provide source authentication (e.g. that the credentials match what other authoritative sources like DNS say) and identity authentication (e.g. that a third party vouches for an entity's identity via issuing a digital certificate).

Example of Risks Involved with Opportunistic Encryption without Authentication

In many ways, an attacker taking advantage of opportunistic encryption without authentication is similar to an attacker subverting a Certificate Authority (CA) to issue a certificate for a given domain to the attacker. This type of attack has been documented several times in the past [2][3] and was successful to an extent in capturing users' private information.

In the above case, an attacker subverted the certificate validation mechanism in the protocol to impersonate a legitimate site. When relying on opportunistic encryption, the attacker does not even need to risk exposure in obtaining a false credential. The attacker can simply generate their own cryptographic key or certificates and insert themselves in the path of the communication. To reduce exposure afterwards, they could act as a Man-in-the-Middle and forward the victim's messages to the legitimate site. When this attack succeeds, the victim is still vulnerable to eavesdropping, as well as more active attacks.

If the protocol specification calls for the use of opportunistic encryption and there are no end user options to require (or signal) authentication then the end users can develop a false sense of security. Non-technical savvy end users who do not understand the differences between confidentiality and authentication may believe they are protected when they are not, and expose private information to a potential attacker.

Dangers of Ignoring Authentication in Protocol Specifications and Deployments

There is a risk when simply specifying a means to add opportunistic encryption to a protocol that an implementer may interpret that authentication is not needed or that authentication will never be desired. As a result, there is a risk that code to perform authentication (e.g. certificate validation) will not be added to implementations, as it would never be needed.

An example is most current real world deployments of SMTP over TLS. Most SMTP servers do not have a set of root certificates installed and therefore do not perform any authentication on the certificate presented. There is an effort to provide a way for a SMTP server to authenticate the certificate using Domain Name System (DNS)-Based Authentication of Named Entities (DANE) [4]. Using DANE, a client can verify the certificate presented by the server during the TLS handshake by looking in the DNS. Here, the domain owner provides a level of authentication of the presented certificate.

Another example is the recommendation to use WiFi security technologies like WiFi Protected Access (WPA or WPA2) to protect against eavesdroppers and browser attacks as seen with Firesheep [5]. However, WPA/WPA2 will only cover the immediate connection and nothing beyond the first hop. An eavesdropper could defeat WPA [6] and eavesdrop on the communication, or monitor the link after the first hop. The use of TLS for web browsing provides superior protection and when coupled with the use of DANE to publish certificate credentials (see below), provides authentication of both source and, depending on how deployed, identity to the end user.

Use of Trust Infrastructures

Providing authentication requires more work (on all parties involved in communication) than opportunistic encryption, but the wheel does not have to be reinvented for every protocol. There are tools and resources available to protocol designers to add authentication support to protocols.

The most ubiquitous is the Domain Name System (DNS), which can be used to store certificate information using the new Resource Record Types (RRTypes) defined by the IETF DANE working group [7]. A client that understands how to query for DANE RRTypes can validate that the certificate presented during a TLS handshake matches what the authoritative domain holder claims, but could also confirm the CA used to obtain the certificate (if a CA issued it), or the local trust anchor used for the domain.

There are other options available for those that may not wish to rely on the existing CA/Public Key Infrastructure (PKI) for certificates. Examples such as the Certificate Transparency [8] work and Sovereign keys [9] use publicly available logs to build trust rather than just assume trust based on receiving any certificate signed by a public CA. The concept is that since the certificate (and holder) is listed on several publicly visible third party services attackers using spoofed certificates would be detectable by clients.

All of these solutions come with their own drawbacks, as well as all requiring more work (and time) being spent by the client in performing authentication. This would likely affect user experience and response time. New user education may also be

needed to help users understand client configuration options that may be available as well as authentication results in protocols.

Conclusions

Encouraging the use of confidentiality in Internet communication will benefit the end user. Standards bodies like the IETF are correct in identifying that passive monitoring (pervasive or targeted) is an attack that protocols should protect against. Opportunistic encryption can and should be used as a last resort to provide a minimal level of confidentiality to protect end users' privacy. Ideally, protocols should be specified to allow communicating parties to authenticate each other, as opportunistic encryption may provide a false sense of security in naive end users.

Since opportunistic encryption usually entails no authentication, end users may believe their privacy is protected when in fact they are sending data to an imposter. Protocol designers should consider authentication as important as confidentiality since the designers cannot always determine when an end user would desire authentication.

References

[1] "We Will Strengthen the Internet" IETF blog Nov 2013 http://www.ietf.org/blog/2013/11/we-will-strengthen-the-internet/

[2] Microsoft Security Advisory 2607712 Fraudulent Digital Certificates Could Allow Spoofing. Microsoft Corp. Aug 2011. http://technet.microsoft.com/en-us/security/advisory/2607712

[3] " Microsoft, Yahoo, Google, Skype, Mozilla Sites Hit by Fraudulent Certificates". eWeek. March 2011. http://www.eweek.com/c/a/Security/Microsoft-Yahoo-Google-Skype-Mozilla-Sites-Hit-by-Fraudulent-Certificates-619996/

[4] V. Dukhovni, W. H. Hardakar. "SMTP security via opportunistic DANE TLS" Work in Progress. http://datatracker.ietf.org/doc/draft-ietf-dane-smtp-with-dane/

[5] Firesheep http://codebutler.com/firesheep/

[6] V. Kumkar, A. Tiwari, P. Tiwari, A. Gupta, and S. Shrawne. "Vulnerabilities of Wireless Security protocols (WEP and WPA2)". International Journal of Advanced Research in Computer Engineering & Technology. Volume 1, Issue 2, April 2012.

[7] DANE Working Group Charter http://datatracker.ietf.org/wg/dane/

[8] Certificate Transparency project homepage http://www.certificate-transparency.org/

[9] Sovereign Keys project homepage https://www.eff.org/sovereign-keys